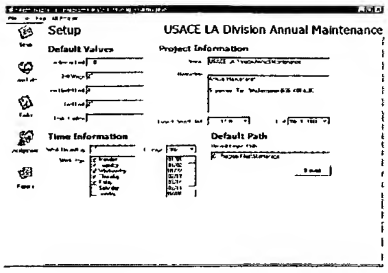
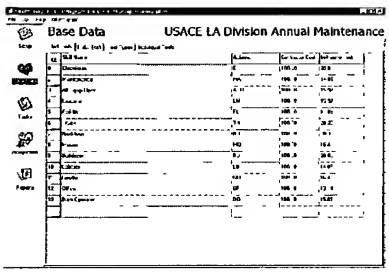
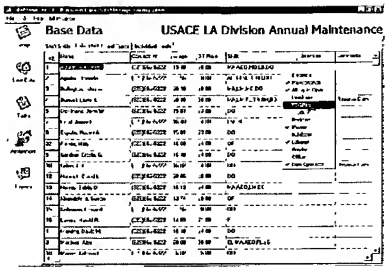
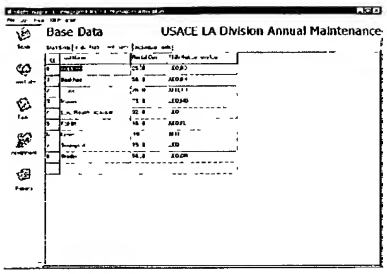
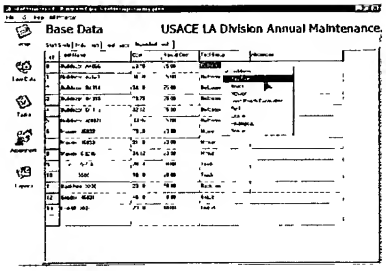
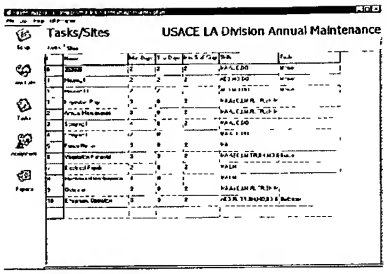

Setup
USACE LA Division Annual Maintenance
Default Values
Project Information
Time Information
Default Path

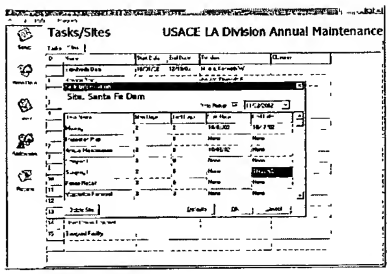

Base Data
USACE LA Division Annual Maintenance

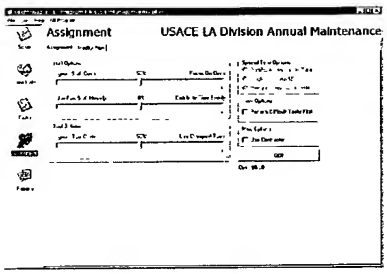

Base Data
USACE LA Division Annual Maintenance

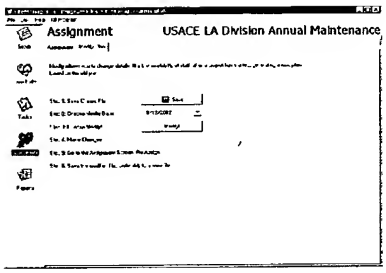

Base Data
USACE LA Division Annual Maintenance

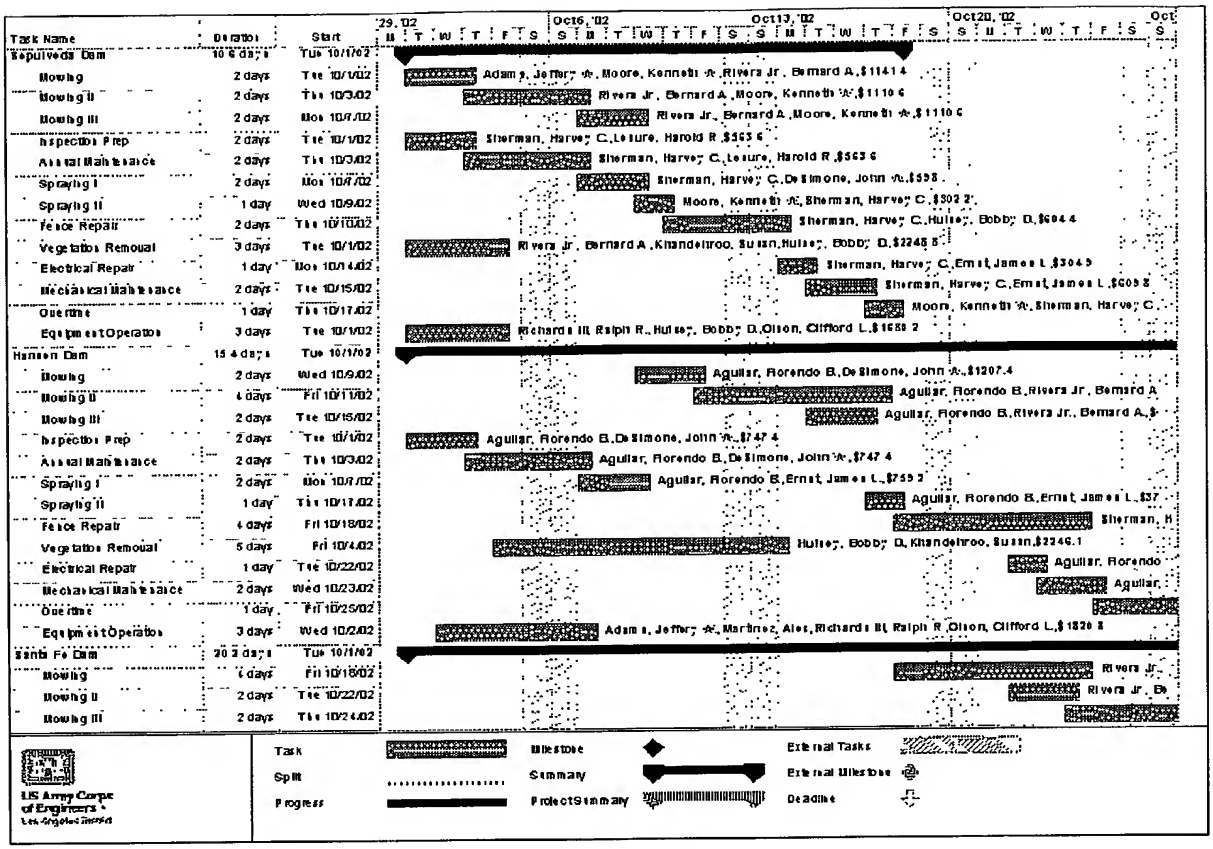

Base Data
USACE LA Division Annual Maintenance

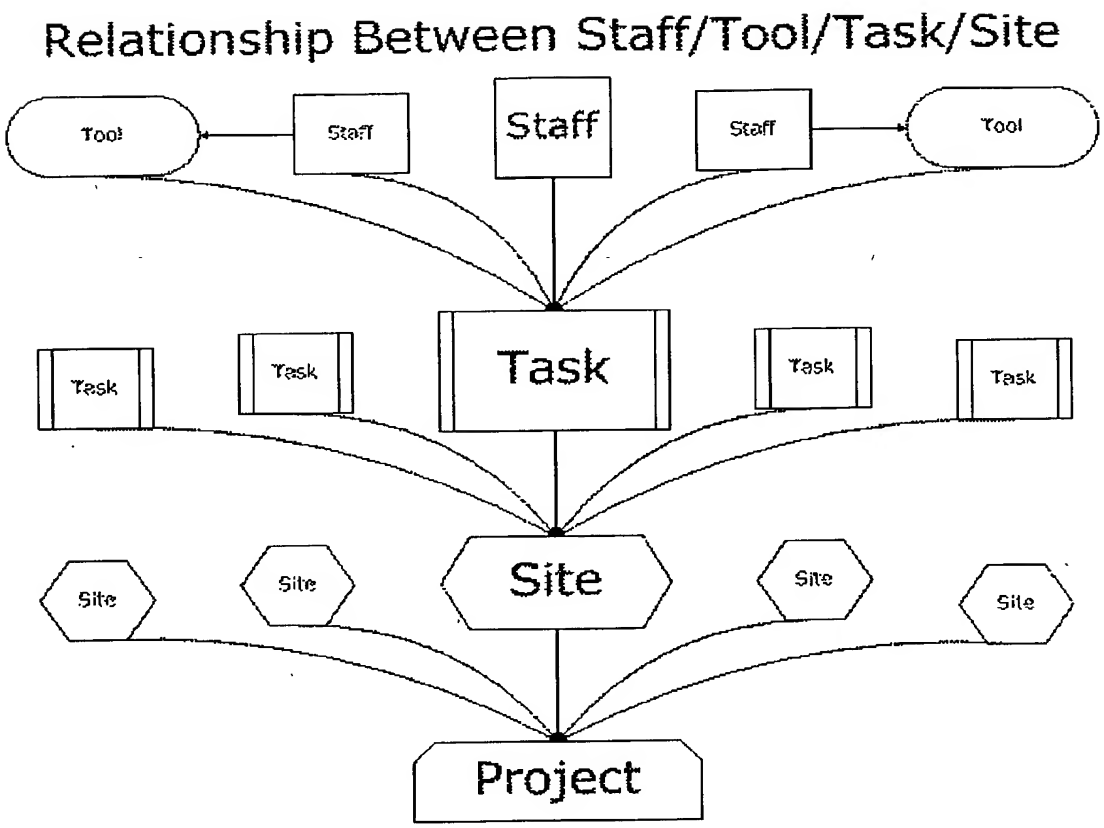

Tasks/Sites
USACE LA Division Annual Maintenance


Tasks/Sites
USACE LA Division Annual Maintenance
Site, Santa Fe Dam


Assignment
USACE LA Division Annual Maintenance


Assignment
USACE LA Division Annual Maintenance


Task Name
Duration
Start
Sepulveda Dam 10 6 days Tue 10/1/02
Mowing 2 days Tue 10/1/02
Mowing II 2 days Thu 10/3/02
Mowing III 2 days Mon 10/7/02
Inspection Prep 2 days Tue 10/1/02
Annual Maintenance 2 days Thu 10/3/02
Spraying I 2 days Mon 10/7/02
Spraying II 1 day Wed 10/9/02
Fence Repair 2 days Thu 10/10/02
Vegetation Removal 3 days Tue 10/1/02
Electrical Repair 1 day Mon 10/14/02
Mechanical Maintenance 2 days Tue 10/15/02
Overtime 1 day Thu 10/17/02
Equipment Operation 3 days Tue 10/1/02
Hansen Dam 15 4 days Tue 10/1/02
Mowing 2 days Wed 10/9/02
Mowing II 4 days Fri 10/11/02
Mowing III 2 days Tue 10/15/02
Inspection Prep 2 days Tue 10/1/02
Annual Maintenance 2 days Thu 10/3/02
Spraying I 2 days Mon 10/7/02
Spraying II 1 day Thu 10/17/02
Fence Repair 4 days Fri 10/18/02
Vegetation Removal 5 days Fri 10/4/02
Electrical Repair 1 day Tue 10/22/02
Mechanical Maintenance 2 days Wed 10/23/02
Overtime 1 day Fri 10/25/02
Equipment Operation 3 days Wed 10/2/02
Santa Fe Dam 20 2 days Tue 10/1/02
Mowing 4 days Fri 10/18/02
Mowing II 2 days Tue 10/22/02
Mowing III 2 days Thu 10/24/02
29, '02
Oct 6, '02
Oct 13, '02
Oct 20, '02
Oct
Adams, Jeffery A,Moore, Kenneth A,Rivera Jr., Bernard A,$1141 4
Rivera Jr., Bernard A,Moore, Kenneth A,$1110 6
Rivera Jr., Bernard A,Moore, Kenneth A,$1110 6
Sherman, Harvey C,Leisure, Harold R,$563 6
Sherman, Harvey C,Leisure, Harold R,$563 6
Sherman, Harvey C,DeSimone, John A,$558
Moore, Kenneth A,Sherman, Harvey C,$302 2
Sherman, Harvey C,Hulsey, Bobby D,$604 4
Rivera Jr., Bernard A,Khandelwal, Susan,Hulsey, Bobby D,$2245 8
Sherman, Harvey C,Ernst, James L,$304 9
Sherman, Harvey C,Ernst, James L,$609 8
Moore, Kenneth A,Sherman, Harvey C,
Richards III, Ralph R,Hulsey, Bobby D,Olson, Clifford L,$1658 2
Aguilar, Rorendo B,DeSimone, John A,$1207 4
Aguilar, Rorendo B,Rivera Jr., Bernard A
Aguilar, Rorendo B,Rivera Jr., Bernard A,$
Aguilar, Rorendo B,DeSimone, John A,$747 4
Aguilar, Rorendo B,DeSimone, John A,$747 4
Aguilar, Rorendo B,Ernst, James L,$759 2
Aguilar, Rorendo B,Ernst, James L,$37
Sherman, H
Hulsey, Bobby D,Khandelwal, Susan,$2246 1
Aguilar, Rorendo
Aguilar,
Adams, Jeffery A,Martinez, Alex,Richards III, Ralph R,Olson, Clifford L,$1820 8
Rivera Jr.,
Rivera Jr., B
US Army Corps of Engineers
Los Angeles District
Task
Split
Progress
Milestone
Summary
Project Summary
External Tasks
External Milestone
Deadline


Relationship Between Staff/Tool/Task/Site
Tool
Staff
Staff
Staff
Tool
Task
Task
Task
Task
Task
Site
Site
Site
Site
Site
Project

```
// This class allows for sorting based on scores
      // without losing track of the original object
      class ObjScore
      {
      public:
            int id;
            int score;
      }

      void GetStaffList(Task* t, Array<Personel*> &list)
      {
            // Get All the Staff who can do this task
            // and put it in the array "list"
            GetAllSkilledStaff(t, list)

            // Make an array of score objects
            // the size of "list"
            Array<ObjScore> scores(list.length())

            for(i=0;i<list.length();i++)
            {
                  // Give this person's ID to the score object
                  scores[i].id = list[i]->id

                  // Make this score affected by the staff's cost
                  // It is negative to show that the higher the cost,
                  // the worse the score. Multiply it by some factor
                  // based on the specified criteria
                  scores[i].score = -(factor) * list[i]->cost

                  // Make this score affected by the number of days
                  // worked by the staff already. This is multiplied
                  // by a work-distribution factor. If the user chooses
                  // to spread work evenly, then the less-worked staff
                  // will be given a higher score
                  scores[i].score += (factor) * DaysWorked(list[i])
            }

            // Sort the list based on score. This uses the quicksort
            // sorting method, with modifications to take a list of
            // ObjScore and not numbers

            OSQuickSort(os,0,os.length()-1)

            // Now make the list reflect the newly ordered list of
            // scores
            for(i=0;i<list.length();i++)
                  list[i] = GetStaffFromID(scores[i].id)
      }
```


[redacted]

```
// Shows how these resources are inter-connected

// Basic Resource Category
class Resource

public:
        int id;                                 // ID number to identify resource
        TString name;                           // Name of Tool/Staff
      ⇨ TArray<ResTask*> tasks;                 // List of tasks and days
        float cost;                             // Base Rate
        bool useresc;                           // Is this resource available
        TArray<Absence> abs;                    // List of absence Dates

        Resource();
 ;

// Tool Resource
class Tool : public Resource

public:
        TArray<int> toolIdxs;                   // List of tool categories this
                                                // tool falls into
 ;

// Staff Resource
class Personel : public Resource

public:
        TArray<int> skillno;                    // Skills this staff has
        TString contact,comment;                // Contact info and comments
      ⇨ TArray<ResPend*> tendering;             // Which site/dates this staff
                                                // tenders for
        float otrate;                           // Overtime Rate

        Personel();
 ;

class Site

public:
        TString name;                           // Name of Site
        int id;                                 // Site ID (For searching/saving)
      ⇨ TArray<Group*> groups;                  // Groups (of Tools and Staff) that
                                                // can work at this site
        TArray<Personel*> tenders;              // Possible Site Tenders
      ⇨ TArray<Task*> tasks;                    // Tasks occurring at this site
        TArray<Site*> sub;                      // Sub-sites of this site
      ⇨ TArray<TaskPers*> curtender;            // Tenders and Dates of Tendering

        bool closed;                            // Is this site closed down forever
        TArray<Absence> abs;                    // Site closures

      ⇨ TArray<SiteTask*> st;                   // Site/Task Linkage

    Site();
 ;

class Task

public:
      ⇨ SiteTask* st;                           // Site/Task Linkage
      ⇨ Site* site;                             // Site where this Task takes place
      ⇨ TArray<TaskPers*> personel;             // List of staff/days working on
      ⇨ TArray<TaskPers*> ot;                   // List of overtime staff
      ⇨ TArray<TaskTool*> tools;                // List of tools/days
        int start,end;                          // Task's start and end dates

        Task();
 ;
```